

# RESCON: Educational Project Scheduling Software

FILIP DEBLAERE, ERIK DEMEULEMEESTER, WILLY HERROELEN

*Department of Decision Sciences and Information Management, Katholieke Universiteit Leuven, Naamsestraat 69, Leuven 3000, Belgium*

Received 14 August 2008; accepted 2 December 2008

**ABSTRACT:** In this article we discuss a freely downloadable educational software tool for illustrating project scheduling and project management concepts. The tool features exact and heuristic scheduling procedures and visualizes project networks, project schedules, resource profiles, activity slacks, and project duration distributions. © 2009 Wiley Periodicals, Inc. *Comput Appl Eng Educ*; Published online in Wiley InterScience (www.interscience.wiley.com); DOI 10.1002/cae.20314

**Keywords:** project scheduling; project management; educational software; visualization; scheduling algorithms

---

## INTRODUCTION

Project management competence is of the essence in many industries. Indeed, be it a software development project, a shut-down and maintenance project, an R&D project or any other industrial project: the careful planning, scheduling, and management of the project is as important as the project content itself. In this article we describe freely downloadable<sup>1</sup> educational project scheduling software called RESCON (from RESource CONstrained) that is used at the Katholieke Universiteit Leuven (Belgium) for the

illustration of project management concepts. Some recent studies demonstrate the effectiveness of educational software tools in the field of operations management [1,2]. The RESCON software mainly focuses on the scheduling part of the project management field. In other words, we assume that the project content has been defined, the different project activities have been identified and precedence relations have been introduced between the activities. Furthermore, we assume that the required amount of resources for the execution of every individual activity has been calculated and the activity durations have been estimated. This information, along with the limited availability of resources, can then be seen as an instance of the resource-constrained project scheduling problem (RCPSP) [3], and can be solved with one of the many types of scheduling algorithms, some of which are embedded in the software.

---

Correspondence to: F. Deblaere  
(filip.deblaere@econ.kuleuven.be).

© 2009 Wiley Periodicals Inc.

<sup>1</sup>RESCON can be downloaded from <http://www.econ.kuleuven.be/rescon/>.

The basic RCPSP assumes that activity durations are deterministic and known to the project manager. In practice, however, activity durations can only be estimated and the actual activity durations may deviate from these estimates. These deviations may have many causes, such as underestimation of the work content, errors during execution, the unavailability of key staff or equipment, etcetera. Effective project management requires that the project manager takes these risks into account. This is why it is advisable to perform a risk analysis on the activities, leading to activity duration distributions rather than deterministic activity durations. Using this information, the project duration distribution can be calculated using Monte Carlo simulation and proactive scheduling steps can be taken to make the project schedule more robust. All of this (except for the risk analysis itself) is also supported in the software.

The remainder of this article is organized as follows. In the next section we give an introduction to the basic RCPSP and we discuss the corresponding features that are implemented in the software. The third section then elaborates on the software features with respect to project scheduling under activity duration uncertainty. In the fourth section we comment on the student feedback concerning the educational value of the software. Finally, we end our discussion with some concluding remarks.

## THE BASIC RCPSP

The basic RCPSP involves a project network  $G(N, A)$  with a set  $N$  of nodes representing the project activities. The activities in the network are subject to so-called zero-lag finish-start precedence constraints  $(i, j) \in A$ , indicated by the arcs of the network. The presence of such an arc implies that an activity  $j$  cannot start before its predecessor activity  $i$  has finished. We assume the presence of  $m$  renewable resource types, with a per period availability  $a_k$ ,  $k \in K$  with  $K = \{1, \dots, m\}$ . These resource types can be, for example, manpower or machinery, and are assumed to have a constant availability per time period throughout the entire project horizon. The resources are *renewable* in the sense that even if we “use” the resources during a certain time period  $t$ , they will again be available at full capacity during every later time period  $t + 1, t + 2, \dots$ . The project activities  $i \in N$  require an integer per period amount  $r_{ik}$  of resource type  $k$ ,  $k \in K$ . A solution to the RCPSP then consists of a vector of start times  $s_i$ ,  $i \in N$ , such that the resource and precedence constraints are satisfied, and the project makespan is minimized.

In Figure 1, the project network window of the RESCON software is shown. Each rectangle corresponds with an activity and shows the activity number, the deterministic activity duration and the

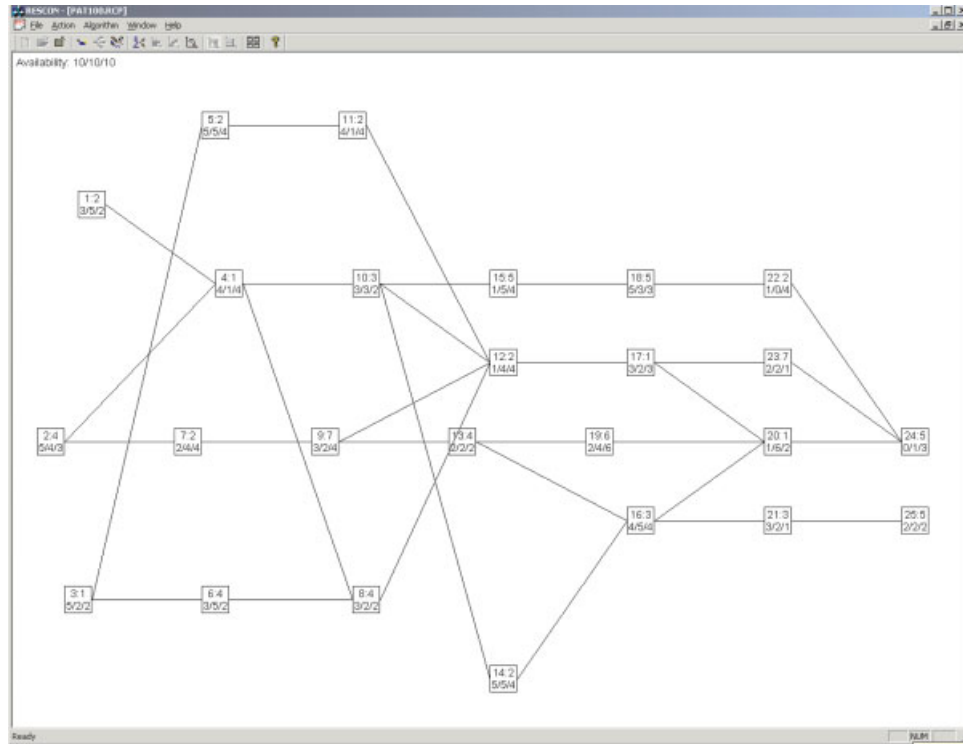
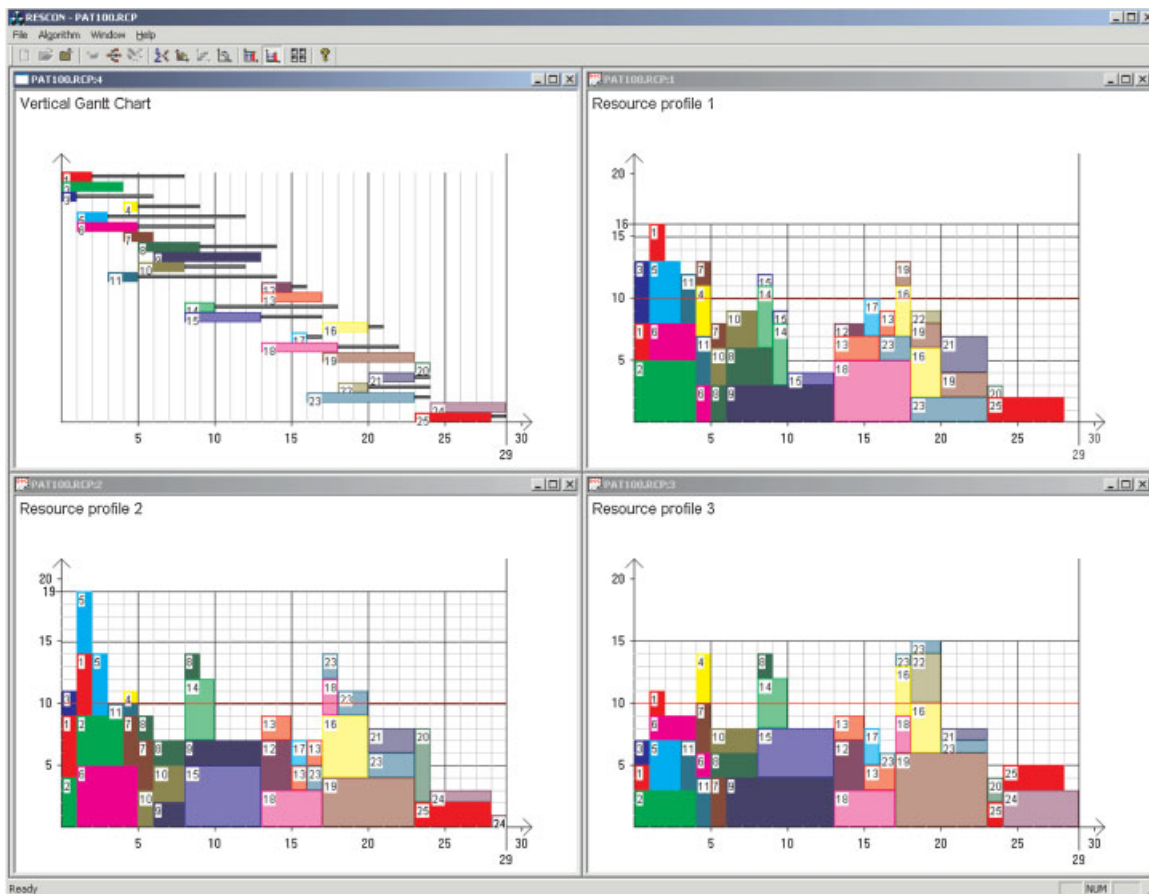


Figure 1 Activity network window.

requirements for the different renewable resource types. For instance, the activity in the upper left corner is activity 1 and has a duration  $d_1$  equal to two time units. It requires three units of the first resource type ( $r_{11}=3$ ), five units of the second resource type ( $r_{12}=5$ ), and two units of the third resource type ( $r_{13}=2$ ). The per period availability  $a_k$  of the three resource types is indicated in the top left corner of the project network window. Standard operations such as adding or removing activities, introducing or removing precedence relations and changing activity properties can be dealt with very easily through the user interface. As for loading a project into the software, project files in the “rcp” format (this is the format used in the project scheduling library PSPLIB [4] and in the RanGen network generator [5]) can be read in. Alternatively, the user can build up a project network from scratch, starting with an empty project and adding activities and precedence relations one at a time. These networks can then be exported to a file in the aforementioned “rcp” format.

The most basic temporal analysis that can be performed on an RCPSP instance is the so-called critical path analysis. This analysis involves the calculation of the earliest possible start times of all activities while ignoring the renewable resource constraints, and provides valuable initial insights in terms of the minimal project makespan and the peak resource consumption. The software allows for the calculation of this so-called *early start schedule* and the visualization of this schedule by means of resource profiles and a vertical Gantt chart. A screenshot of this feature is shown in Figure 2. As can be seen from the figure, the resource profiles and the Gantt chart are shown in child windows of the RESCON main window. The user can then tile or rearrange these windows to get the desired overview of the schedule.

In the top left window, the vertical Gantt chart is shown. The colored rectangles indicate the time slots where the activities are executed, while the thin gray bars extending to the right represent the activity slacks, that is, the amount of time the finish of the respective activities can be delayed without increasing



**Figure 2** Early start schedule. [Color figure can be viewed in the online issue, which is available at [www.interscience.wiley.com](http://www.interscience.wiley.com).]

the minimal obtainable makespan. The other three windows are the resource profiles, one for every resource type. They indicate the resource consumption (on the vertical axis) of the different activities per time unit (time is shown on the horizontal axis). For instance, when looking at resource profile 1, we see that activity 18 starts at time instant 13 and finishes at time instant 18. During its execution, it uses five units of the first renewable resource type. In all four windows, rectangles corresponding to a certain activity will be drawn in the same color. The activity number is shown in the top left corner of the rectangles. The key observations to be made from this analysis are the lower bound on the project makespan being equal to 29 time units, and the fact that the resource constraints are violated for all three resource types (the maximum capacity being indicated by the horizontal red line).

In most projects, resource constraints will be binding and as a consequence, the above schedule will not be executable. Therefore, project management must resort to scheduling strategies that result in resource feasible schedules with (preferably) a short project makespan. The scheduling algorithms for the RCPSP described in the literature can be roughly divided into three categories, all of which are represented in the software tool:

- *List scheduling algorithms*: Given a priority list of the project activities, the serial scheduling generation scheme [6] or the parallel schedule generation scheme [7] can be used to generate a schedule. These constructive heuristics are very fast in terms of computation time but can potentially yield schedules with a makespan high above the optimal value.
- *Exact procedures*: The most effective exact procedures for solving the RCPSP are dedicated branch-and-bound procedures [8]. Due to the complexity of the problem, the computation time of these procedures increases rapidly when the number of activities increases. Instances with up to 30 activities can be solved within reasonable time limits using dedicated branch-and-bound procedures. Larger instances may take a prohibitive amount of time to solve, so that one has to resort to heuristic methods.
- *Metaheuristics*: These algorithms perform a broad search in the solution space in order to find a good solution for the RCPSP. They often use priority lists (and list scheduling) as a subroutine to generate intermediate schedules. Some recent procedures can be found in Refs. [9–13].

The software has the ability to produce schedules using a variety of procedures. The presence of these different methods is important in that it enables students to experience the power of the more advanced methods when compared to simple heuristics. The algorithms embedded in the software tool are the following:

- *Simple list scheduling*: The software features the serial as well as the parallel schedule generation scheme, both of which must be used in combination with a priority list. Eight popular priority rules are provided, including Minimum Activity Slack (MINSLK), Late Finish Time (LFT) and Late Start Time (LST). The different schedule generation schemes and priority lists can then be used in combination with forward planning, backward planning, and bidirectional planning. These planning directions dictate the order in which the priority list is scanned and the schedule is built up. In total, this results in 48 different list scheduling heuristics.
- *Exact branch-and-bound*: A variant of the exact branch-and-bound procedure proposed by Demeulemeester and Herroelen [8] is embedded in the software. The exact procedure is able to solve project networks with any number of activities and any number of resource types. Of course, for large and difficult instances, the calculations may take very long.
- *Tabu search*: A rather basic tabu search procedure is included to allow for the calculation of compact project schedules for arbitrary large or difficult instances. The tabu search relies on list scheduling and includes a simple diversification scheme. To allow for optimal flexibility concerning computation time and solution quality, a time limit can be supplied by the user.
- *User defined algorithms*: The software allows for the integration of third-party schedule generation code into the software. More specifically, support is provided to link with an external Dynamic Link Library (DLL) containing the implementation of a function with a pre-defined signature that generates a project schedule based on the project data that is passed as an argument to the function. This way, students or researchers can write their own project scheduling code and test it directly through the user interface. The functional requirements for the DLL plus a fully functional example of DLL project scheduling source code are made available on the website of the software.

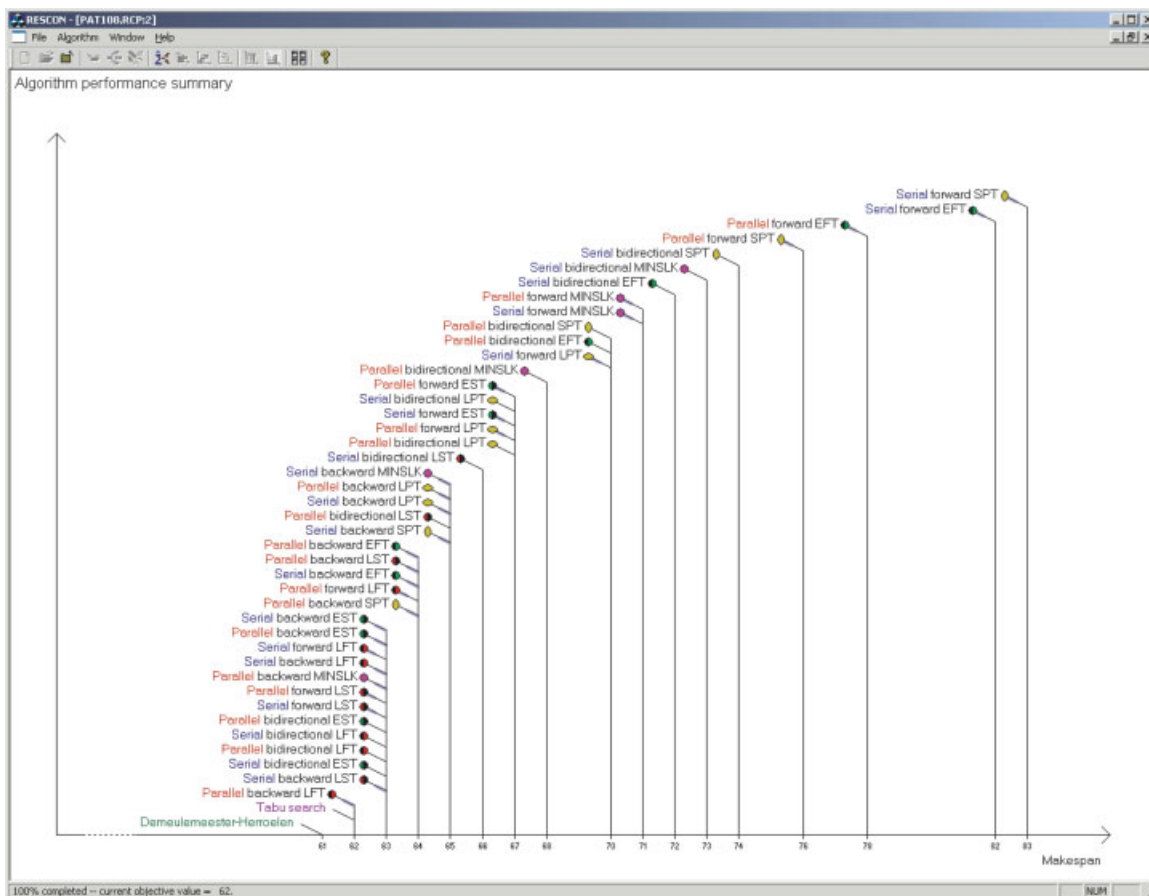
As a final feature regarding these scheduling algorithms, the software includes a window where the performance of all of the scheduling algorithms with respect to the obtained project makespan is summarized on a single time line. This way, the large differences in performance between the different algorithms on the same project instance are made clear. A screenshot of this feature is shown in Figure 3. The optimal procedure obtains a makespan of 61 time units whereas the procedure that performs worst (in this case it is the serial forward shortest processing time list scheduling heuristic) obtains a makespan of 83 time units.

### PROJECT SCHEDULING UNDER ACTIVITY DURATION UNCERTAINTY

In project management, there has been a recent trend towards *robust scheduling*. This paradigm is based on the central idea that it is more important to produce a *realistic* schedule than a minimum duration schedule.

Indeed, an optimal schedule is only optimal to the extent that the reality behaves as expected and this latter condition is rarely satisfied. In robust scheduling, one tries to generate a project schedule that takes uncertainty into account and anticipates on the occurrence of unforeseen events. A well-known methodology for coping with uncertainty is the *critical chain/buffer management* approach, proposed by Goldratt [14]. In short, this methodology inserts a number of time buffers into the project schedule, the duration of which is determined using rules of thumb and without taking advantage of certain knowledge concerning activity duration distributions or activity rescheduling flexibility.

As project scheduling under uncertainty is critical in any practical project management context, we have included a robust scheduling procedure in the software. The featured algorithm is called the Starting Time Criticality (STC) heuristic and is described in Ref. [15]. The procedure is more sophisticated than the critical chain approach, in that activity duration distributions and activity-rescheduling flexibility is



**Figure 3** Algorithm performance summary. [Color figure can be viewed in the online issue, which is available at [www.interscience.wiley.com](http://www.interscience.wiley.com).]

taken into account when inserting time buffers into the schedule. In a nutshell, the STC heuristic protects the most inflexible activities from those predecessors in the schedule that have the largest chance of finishing later than planned. A detailed description of the algorithm itself is beyond the scope of this article.

For the integration of this robust scheduling procedure in the software, we need some additional information for each activity. First of all, in addition to the deterministic activity duration, we need a duration distribution per activity. For reasons of clarity and simplicity, we have chosen to use triangular distributions in the software. When the user double-clicks on an activity, the activity properties window pops up and reveals the characteristics of that activity. A screenshot is shown in Figure 4. In this window, the parameters of the triangular distribution can be modified, while a visualization of the corresponding probability density function is updated in real time.

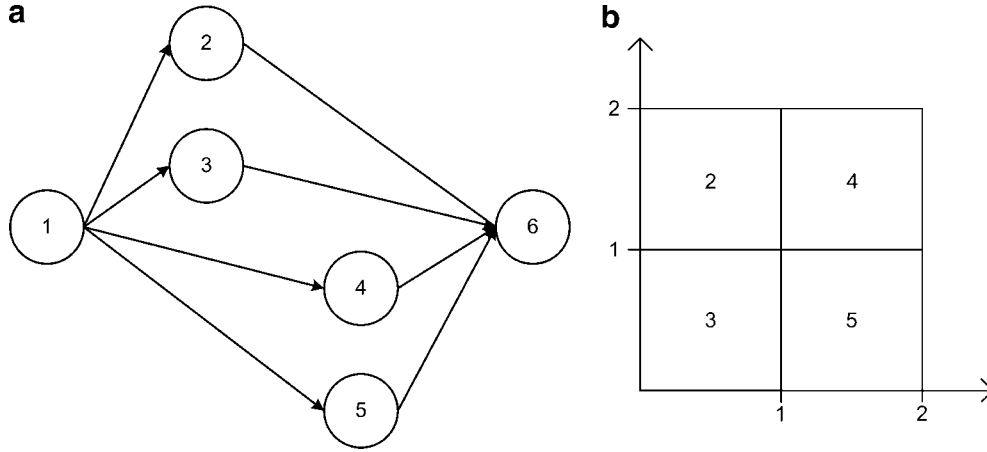
A second parameter required by the STC algorithm is an *inflexibility weight* per activity. A high inflexibility weight reflects the fact that it is highly undesirable that the involved activity would need to be rescheduled due to disruptions during project execution. For instance, if there is an activity in a construction project that needs a capital-intensive

and difficult-to-obtain resource such as a crane, we want a high certainty that this particular activity can start at its projected start time. If we assign a high inflexibility weight to this activity, the STC heuristic will put more effort in protecting this activity than it will in protecting the more flexible activities. The inflexibility weight can also be modified in the same activity properties dialog.

The calculation of a robust schedule using the STC heuristic proceeds in two steps. First, the user must select an algorithm for the generation of an unbuffered schedule. Any of the embedded schedule generation algorithms can be used but in general, more sophisticated algorithms (such as the exact procedure or the tabu search) will yield better results. Given the algorithm of choice, the software will calculate an initial schedule with a makespan equal to  $T$  time units. In a second step, the user must supply a *project deadline*  $T' = T + \Delta$ , with  $\Delta$  a positive number of time units that will be used by the STC procedure to buffer the schedule. The adequate value of  $\Delta$  depends on the preferences of the project manager and the variability of the stochastic activity durations. To assist the user in making an informed choice for the value of  $\Delta$ , the software will calculate a cumulative project duration distribution of the initial (unbuffered) schedule using Monte Carlo simulation. For the calculation of such a project duration distribution, a methodology is needed to simulate the project execution given the initial schedule and a set of stochastic activity durations. More specifically, we require a procedure to calculate realized activity start times such that the resulting schedule remains resource feasible. In what follows, we briefly sketch the idea of the methodology implemented in the software.

The procedure used for calculating schedule realizations relies on the concept of a *resource flow network* [16]. Such a network indicates how every single unit of renewable resource is transferred from one activity (when it finishes) to another (when it starts). A resource flow from an activity  $i$  to an activity  $j$  can then be treated as an additional precedence relation between the involved activities. This way, given a set of stochastic activity durations generated during the Monte Carlo simulation, we can calculate a set of realized activity start times by means of an early start schedule of the original project network augmented with the arcs from the resource flow network. A small example will clarify this. In Figure 5a, a project network is shown with four non-dummy activities, all of which have a unit duration and require one unit of the single renewable resource type. This resource type has a per period

**Figure 4** Activity properties dialog. [Color figure can be viewed in the online issue, which is available at [www.interscience.wiley.com](http://www.interscience.wiley.com).]



**Figure 5** Activity network and schedule. (a) Activity network (b) Resource profile.

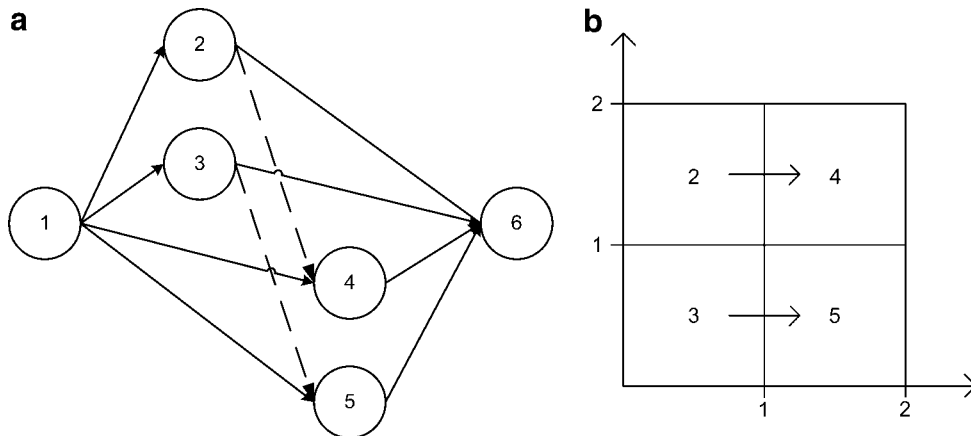
availability of two. A schedule for this project is shown in Figure 5b.

A possible resource flow network for the instance of Figure 5 is shown in Figures 6a,b. The resource flow arcs (the dashed arcs in Figure 6a and the solid arcs in Figure 6b) indicate that when activity 2 finishes, it will pass on its resources to activity 4, when it starts. This results in a precedence relation between activities 2 and 4. The same reasoning holds for activities 3 and 5. When calculating a realized schedule given a set of stochastic activity durations, we will calculate an early start schedule using the network shown in Figure 6a, thus including the additional precedence relations entailed by the resource flows. For instance, if a stochastic duration equal to two time units is generated for activity 2, the precedence relation between activities 2 and 4 will result in the start of activity 4 being delayed until time instant  $t=2$ . The presence of additional precedence

relations caused by resource flows guarantees that an early start schedule is always resource feasible, regardless of realized durations of the activities.

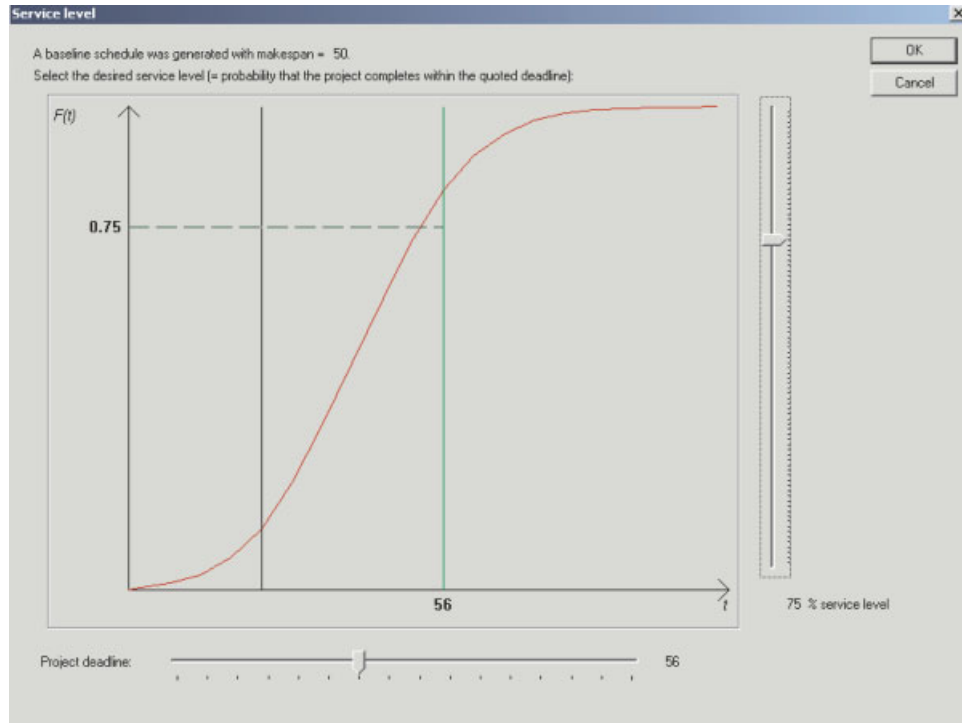
Resource flow networks can be constructed in several ways. In the software we opt for the procedure by Artigues et al. [17]. When the Monte Carlo simulation is complete, the project duration distribution will be shown in a dialog. A screenshot of this is shown in Figure 7.

With every obtainable discrete project makespan now corresponds a *service level*, a probability that the project will finish within this time limit. For the example in Figure 7, a service level of 75% corresponds with a project deadline of 56 time units. Using the two slider bars, the user can learn the makespan corresponding with every service level, and vice versa. For instance, we might be interested in the service level corresponding to the deterministic project duration. In other words, we want to know



**Figure 6** Resource flow network. (a) Activity network with resource arcs. (b) Resource profile with resource flows.





**Figure 7** Project duration distribution. [Color figure can be viewed in the online issue, which is available at [www.interscience.wiley.com](http://www.interscience.wiley.com).]

the probability that the project finishes at its projected finish time, if we were to execute the project using the unbuffered, unprotected version of the schedule that was calculated using the deterministic activity durations. In Figure 7, this deterministic makespan is indicated by the black vertical line. The corresponding probability is equal to 9%. This is already an indication that, assuming the triangular distributions are accurate, a deterministic makespan of 50 time units for this project is highly unrealistic.

The user may now select a desired service level for the project, and the algorithm will subsequently insert buffers in the baseline schedule (i.e., the start of certain activities will be delayed for a certain amount of time), under the constraint that the project is planned to finish at the deadline corresponding to the chosen service level. For the example in Figure 7, if we choose a service level equal to 75%, the algorithm will “inflate” the baseline schedule with the initial makespan of 50 time units to a buffered schedule with a makespan equal to 56 time units. The project manager should then bear in mind that there is still a chance of 25% that the project finishes later than planned.

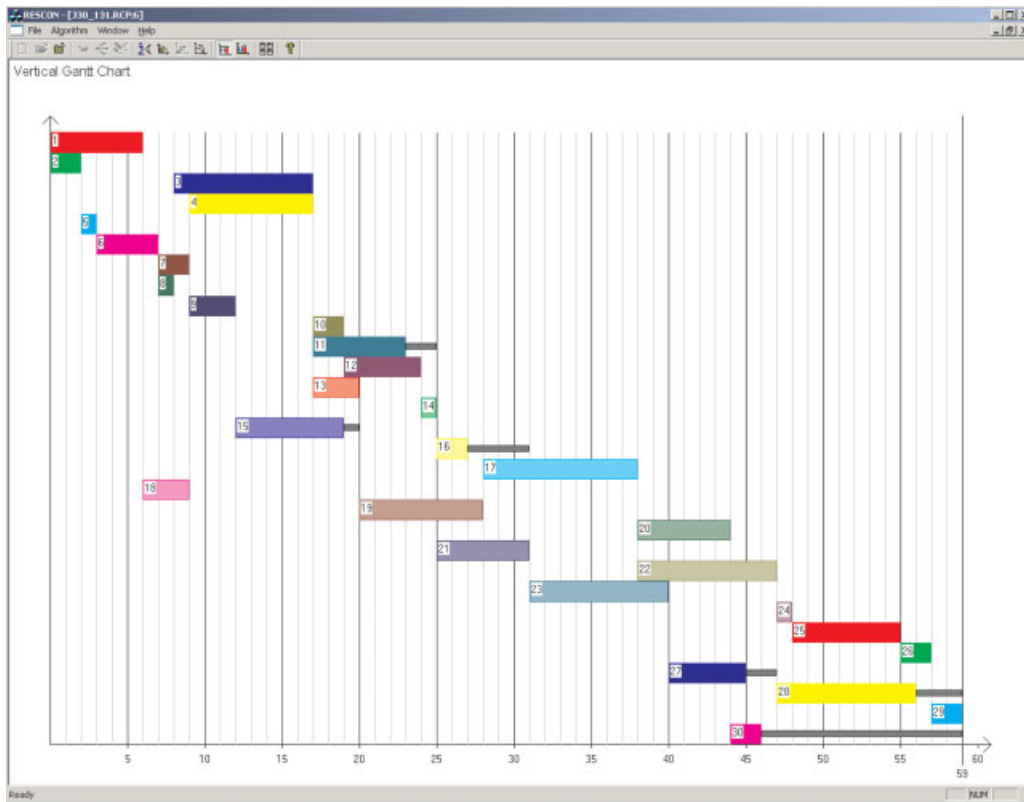
It is enlightening for students (and other users) to visually observe the qualities of a robust schedule. In Figure 8, an unbuffered schedule is generated using a simple list scheduling heuristic resulting in a make-

span of 59 time units. In Figure 9, we apply the STC heuristic to the same project. The initial schedule is generated through the tabu search procedure, and an initial makespan of 50 time units is obtained. This is already nine time units shorter than the schedule generated by the list-scheduling algorithm. Then, we instruct the STC procedure to buffer the schedule until we obtain the same makespan of 59 time units, which will result in a service level of no less than 95%.

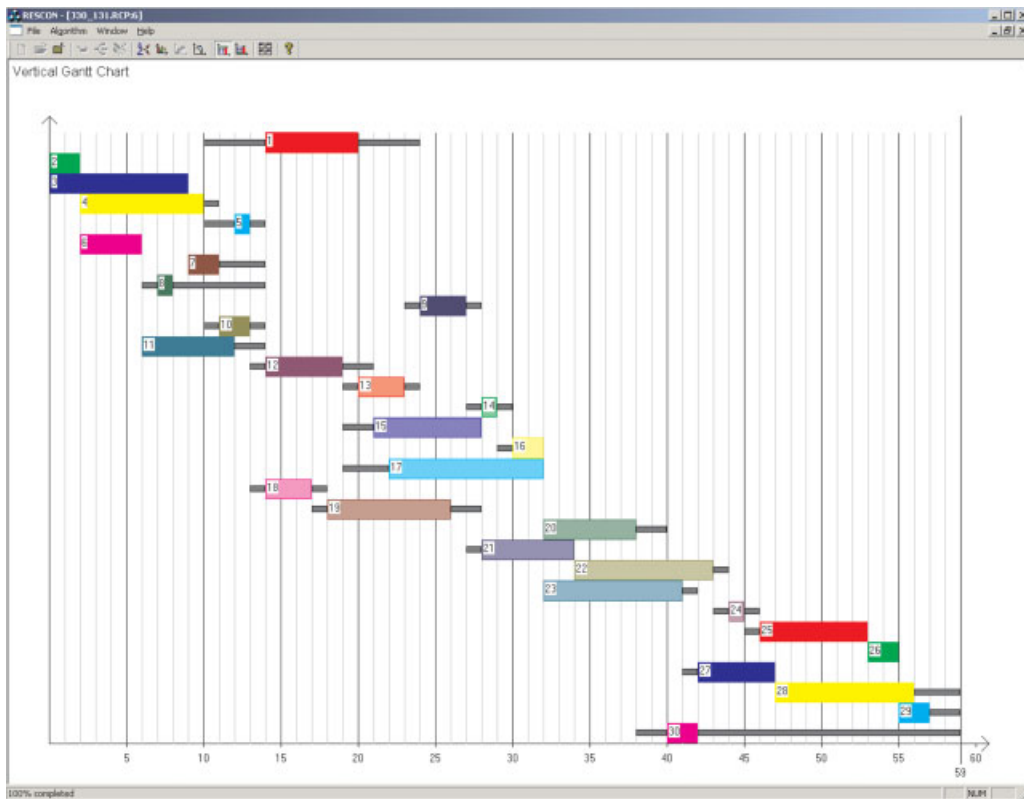
When looking at the Gantt charts corresponding to the different schedules, we can clearly see the difference. In Figure 8, most of the activities have zero slack (recall that slack values are indicated by the gray horizontal bars extending from the activities). This means that they cannot be left- or right-shifted in the schedule without causing a violation of the resource or precedence constraints. Consequently, if the finish time of such an activity is delayed (e.g., due to an underestimation of the work content), at least one of the successor activities in the schedule will be delayed as well, which is bad for the robustness of the schedule. By contrast, in Figure 9 the vast majority of the activities have strictly positive slack values. Moreover, the activities with a high inflexibility will have been given adequate protection, as a result of the working logic of the STC heuristic.

In summary, two important ideas can be conveyed through this analysis. First of all, good scheduling





**Figure 8** Unbuffered schedule obtained through simple list scheduling. [Color figure can be viewed in the online issue, which is available at [www.interscience.wiley.com](http://www.interscience.wiley.com).]



**Figure 9** Buffered schedule using the STC heuristic. [Color figure can be viewed in the online issue, which is available at [www.interscience.wiley.com](http://www.interscience.wiley.com).]

algorithms will allow for a more efficient usage of the available resources and hence result in shorter schedules. Secondly, if we buffer these short initial schedules using a well-designed buffering procedure, we will be able to obtain schedules with a service level that is potentially a lot higher than the service level corresponding to unbuffered schedules obtained through simple planning methods such as list scheduling, while the project makespan remains the same. Moreover, activities in the buffered schedule will have higher slack values, and the schedule can be constructed in such a way that the most inflexible activities receive the best protection against disruptions.

## STUDENT FEEDBACK

The educational software is currently used at Katholieke Universiteit Leuven (Belgium) in the one semester Project and production scheduling course attended by 121 Master students in commercial engineering, taking a major or minor in production management and logistics. A major part of the course essentially deals with fundamental scheduling problems in a project environment (resource leveling and resource-constrained project scheduling) and a manufacturing floor setting (single machine and parallel machine scheduling, flow shop, and job shop). The tool is extensively used in the classroom for studying the fundamental interaction between time (activity durations and lead times), routing and precedence relations, resource requirements and resource availability. Students also use the tool in preparing the classroom and home assignments. The software has also been used in executive seminars dealing with project management in general and project scheduling in particular.

The following observations came from course evaluations and conversations with course and seminar participants:

- Students appreciate the user-friendly features of the software: easy entering of the problem data, graphical representation of the activity network bar chart and resource profiles.
- Students appreciate the possibilities offered by the tool for gaining insight in the time-precedence-resource interplay and the possible misconceptions about project scheduling and time buffering in a resource-constrained environment. Especially the what-if analysis capabilities of the software are highly appreciated: immediate visualization of the impact of changes in

project structure, activity durations, precedence relations, resource requirements, and resource availabilities.

- Students highly appreciate the possibility offered by the software to confront the quality of the schedules obtained by the built-in exact and heuristic solution procedures with the schedules generated by commercial planning software (e.g., Microsoft Project<sup>®</sup>). The fact that schedule quality may very widely differ among the procedures and the fact that the schedules generated by commercial software may be far off the optimum add to their understanding that effective project and machine scheduling really matters.
- Students appreciate the built-in methodology for generating proactive (time-buffered) baseline schedules that are adequately protected against anticipated disruptions. Especially the interaction between timely project completion probability (service level) and due date setting adds to their understanding, as well as the possibility to confront the stability (solution robustness) and makespan performance (quality robustness) of the generated schedules with the schedule plans generated by commercial critical chain based software (e.g., ProChain<sup>®</sup>).

## CONCLUSIONS

In this article, we have discussed a piece of educational software for the illustration of project management concepts. The software visualizes activity networks, Gantt charts, resource profiles, early and late start schedules, and project duration distributions. For the generation of resource feasible schedules, a variety of list scheduling heuristics are embedded in the software (48 in total). High quality schedules (in terms of project duration) can be generated through an exact branch-and-bound procedure or a heuristic tabu search procedure.

Recently, risk management and adequate project planning and scheduling under uncertainty have been identified as critical conditions for successful project management. Therefore, the feature set of the software includes project scheduling under activity duration uncertainty. Using triangular activity duration distributions, the software allows for the generation of cumulative distribution functions of the project duration. Also, the stochastic activity durations and the activity inflexibility weights facilitate the generation of buffered schedules with an arbitrarily large service level, that include the

necessary protection of individual activities against the largest threats posed by activity duration uncertainty. The combination of state-of-the-art scheduling algorithms with an elaborate schedule visualization component results in a valuable tool for illustrating advanced project management concepts in a classroom setting. The educational value of the tool is confirmed by the students, who greatly appreciate the tool's ability to visualize some critical focal points in the field of project management and scheduling.

## REFERENCES

- [1] S. Reynard, O. Gomis-Bellmunt, A. Sudrià-Andreu, O. Boix-Aragonès, and I. Benítez-Pina, Flexible manufacturing Cell SCADA systems for educational purposes, *Comput Appl Eng Educ* 16 (2008), 21–30.
- [2] S. Hsieh, Problem-solving environment for line balancing automated manufacturing systems, *Computer Applications in Engineering Education*, to appear.
- [3] E. Demeulemeester and W. Herroelen, Project scheduling—A research handbook, Vol. 49 of International Series in Operations Research & Management Science, Kluwer Academic Publishers, Boston, 2002, p 203.
- [4] R. Kolisch and A. Sprecher, PSPLIB—A project scheduling library, *Eur J Oper Res* 96 (1997), 205–216.
- [5] E. Demeulemeester, M. Vanhoucke, and W. Herroelen, RanGen: A random network generator for activity-on-the-node networks, *J Scheduling* 6 (2003), 17–38.
- [6] J. Kelley, The critical-path method: Resources planning and scheduling, J Muth and G. Thompson, editors, *Industrial scheduling*, Prentice Hall, Englewood Cliffs, 1963, pp 347–365.
- [7] G. Brooks and C. White, An algorithm for finding optimal or near optimal solutions to the production scheduling problem, *J Ind Eng* 16 (1965), 34–40.
- [8] W. Herroelen, B. De Reyck, and E. Demeulemeester, Resource-constrained project scheduling: A survey of recent developments, *Comput Oper Res* 25 (1998), 279–302.
- [9] J. Alcaraz and C. Maroto, A robust genetic algorithm for resource allocation in project scheduling, *Ann Oper Res* 102 (2001), 83–109.
- [10] S. Hartmann, A self-adapting genetic algorithm for project scheduling under resource constraints, *Nav Res Logistics* 49 (2002), 433–448.
- [11] K. Nonobe and T. Ibaraki, Formulation and tabu search algorithm for the resource constrained project scheduling problem (RCPSP), C. C. Ribeiro and P. Hansen, editors, *Essays and surveys in meta-heuristics*, Kluwer Academic Publishers, Boston, MA, 2002, pp 557–588.
- [12] V. Valls, F. Ballestin, and S. Quintanilla, Justification and RCPSP: A technique that pays, *Eur J Oper Res* 165 (2005), 375–386.
- [13] D. Debels and M. Vanhoucke, A decomposition-based genetic algorithm for the resource-constrained project-scheduling problem, *Oper Res* 55 (2007), 457–469.
- [14] E. Goldratt, *Critical chain*, The North River Press Publishing Corporation, Great Barrington, MA, 1997.
- [15] S. Van de Vonder, E. Demeulemeester, and W. Herroelen, Proactive heuristic procedures for robust project scheduling: An experimental analysis, *Eur J Oper Res* 189 (2008), 723–733.
- [16] C. Artigues and F. Roubellat, A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes, *Eur J Oper Res* 127 (2000), 297–316.
- [17] C. Artigues, P. Michelon, and S. Reusser, Insertion techniques for static and dynamic resource-constrained project scheduling, *Eur J Oper Res* 149 (2003), 249–267.

## BIOGRAPHIES



**Filip Deblaere** is a research assistant and PhD candidate at the Research Center for Operations Management of the Faculty of Business and Economics of K.U. Leuven (Belgium). He earned a Master in Informatics and a Master in Management from K.U. Leuven (Belgium). His research focuses on proactive/reactive project scheduling.



**Erik Demeulemeester** is professor of Operations Management at the Research Center for Operations Management of the Faculty of Business and Economics of K.U. Leuven (Belgium). Dr. Demeulemeester has published extensively in the project scheduling and healthcare management fields. He is co-author of the book *Project Scheduling—A Research Handbook*, published by Springer. His work has appeared in various international academic journals. He is currently associate editor for *Computers and Operations Research*, for the *Journal of Scheduling* and for the *European Journal of Industrial Engineering*. His current research focuses on proactive/reactive project scheduling and health care planning and scheduling.



**Willy Herroelen** is professor of Operations Management at the Research Center for Operations Management of K.U. Leuven (Belgium). Dr. Herroelen has published widely in the project-scheduling field. He is co-author of the book *Project Scheduling—A Research Handbook*, published by Springer. His work has appeared in various international academic journals. He is currently

senior editor for *Production and Operations Management*, area editor for *Computers and Industrial Engineering*, and serves on the editorial board of *International Journal of Production Research*, *Foundation of Computers and Decision Sciences*, and *Journal of Operations and Logistics*. His current research focuses on proactive/reactive resource-constrained project scheduling.